

Rangkuman Matkul PBO Pertemuan 1-6

Nama : Azherino Hadyan P

NPM: 50422308

Kelas : 2IA11

Pertemuan 1-2:

Pengenalan PBO

Pemrograman Berorientasi Objek (OOP) adalah paradigma pemrograman yang berfokus pada objek yang memiliki atribut, perilaku, dan kondisi. OOP pertama kali ditemukan pada tahun 1960 sebagai evolusi dari pemrograman terstruktur, yang memudahkan penulisan program yang kompleks. Metode ini dikembangkan dari bahasa C dan Pascal. Pemrograman berorientasi objek dalam melakukan pemecahan suatu masalah tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Pemrograman berorientasi objek bekerja dengan baik ketika dibarengi dengan Objek-Oriented Analysis And Design Process (OOAD).

PBO

- Menggabungkan fungsi dan data dalam kelas – kelas atau objek – objek
- Memiliki ciri Encapsulation (pengemasan), Inheritance (penurunan sifat) dan Polymorphism (perbedaan bentuk dan perilaku)
- Struktur program ringkas, cukup dengan membuat Objek dan class lalu bekerja berdasarkan object dan class tersebut.
- Kode program sangat re-usable. object dan class dapat digunakan berkali-kali, sehingga dapat menghemat space memori.
- Efektif digunakan untuk menyelesaikan masalah besar, karena OOP terdiri dari class-class yang memisahkan setiap kode program menjadi kelompok - kelompok kecil, sesuai dengan fungsinya
- Sulit diawal (karena harus membuat class) namun selanjutnya akan terasa mudah dan cepat
- Eksekusi lebih cepat karena dieksekusi bersamaan, program hanya mengatur Objek, properties dan method-nya saja

PT(pemograman Terstruktur)

- Memecah program dalam fungsi dan data
- Memiliki ciri Sequence (berurutan), Selection (pemilihan) dan Repetition (perulangan)
- Struktur program rumit karena berupa urutan proses dan fungsi-fungsi
- Re-use kode program kurang
- Efektif digunakan untuk menyelesaikan masalah kecil dan tidak cocok untuk menyelesaikan masalah yang rumit, karena nantinya akan kesulitan menemukan solusi permasalahan ketika terjadi error
- Mudah diawal, namun kompleks diproses selanjutnya
- Eksekusi lebih lambat karena setiap perintah dikerjakan berurutan

Class dalam PBO

Class adalah blueprint atau cetak biru dari objek, yang berfungsi sebagai kerangka dasar. Dalam analogi, class seperti kendaraan dengan ciri-ciri seperti merk, mesin, dan ban, serta kemampuan untuk melakukan tindakan seperti menghidupkan atau mematikan mesin.

Objek pada PBO

Objek adalah entitas yang memiliki atribut, karakter dan kadang kala disertai kondisi. Objek mempresentasikan sesuai kenyataan seperti siswa, mempresentasikan dalam bentuk konsep seperti merek dagang, juga bisa menyatakan visualisasi seperti bentuk huruf (font).

Untuk membuat objek, bisa menggunakan kata kunci new yang digunakan untuk membuat objek baru, selanjutnya menentukan 3 langkah untuk membuat sebuah objek. Yaitu: mendeklarasikan variable, membuat objek baru (Instansiasi) dan pemanggilan konstruktor.

Pertemuan 3-6:

Inheritance

Pewarisan merupakan sebuah bentuk “penggunaan kembali” (*reusability*); dimana **class** baru dibuat dari **class** yang pernah ada yang (biasanya) ditambah fasilitasnya. Setiap **class** turunan dapat menjadi **class** pokok (induk) untuk **class**

turunan yang akan datang. Dalam pewarisan, *constructor* tidak diwariskan pada **class** turunannya, kecuali jika digunakan perintah **super**.

Pewarisan tunggal (single inheritance) adalah pewarisan dari satu class induk. Pewarisan ganda (multiple inheritance) adalah pewarisan dari dua atau lebih class induk, namun Java tidak mendukung pewarisan ganda.

Class yang mewariskan disebut dengan **superclass** / **parent class** / **base class**, sedangkan class yang mewarisi (class yang baru) disebut dengan **subclass** / **child class** / **derived class**. Untuk menerapkan inheritance, gunakan statement “**extends**”.

Keyword “**super**” digunakan oleh subclass untuk memanggil constructor, atribut dan method yang ada pada superclass-nya.

Contoh:

```
super ()  
super (parameter)
```

Contoh untuk memanggil atribut dan method milik superclass-nya :

```
super.namaAtribut  
super.namaMethod(parameter)
```

Encapsulation

Enkapsulasi adalah teknik untuk melindungi properti atau metode tertentu dari suatu kelas agar tidak diakses dan dimodifikasi sembarangan. Perlindungan data ini dilakukan dengan menggunakan access modifiers (hak akses), yang mencakup default, public, protected, dan private.

Accessors

No	Modifier	Pada class dan interface	Pada method dan variabel
1	Default (tidak ada modifier)	Dapat diakses oleh yang sepaket	Diwarisi oleh subkelas dipaket yang sama, dapat diakses oleh method-method yang sepaket
2	Public	Dapat diakses dimanapun	Diwarisi oleh subkelasnya, dapat diakses dimanapun
3	Protected	Tidak bisa diterapkan	Diwarisi oleh subkelasnya, dapat diakses oleh method-method yang sepaket
4	private	Tidak bisa diterapkan	Tidak dapat diakses dimanapun kecuali oleh method-method yang ada dalam kelas itu sendiri

Aksesabilitas	public	private	protected	default
Dari kelas yang sama	Ya	Ya	Ya	Ya
Dari sembarang kelas dalam paket yang sama	Ya	Tidak	Ya	Ya
Dari sembarang kelas di luar paket	Ya	Tidak	Tidak	Tidak
Dari subkelas dalam paket yang sama	Ya	Tidak	Ya	Ya
Dari subkelas di luar paket	Ya	Tidak	Ya	Tidak

Polymorphism

Overloading adalah diperbolehkannya dalam sebuah class memiliki lebih dari satu nama function/method yang sama tetapi memiliki parameter/argument yang berbeda.

```

1 public class Overloading {
2     public void Tampil(){
3         System.out.println("I love Java");
4     }
5     public void Tampil(int i){
6         System.out.println("Method dengan 1 parameter = "+i);
7     }
8     public void Tampil(int i, int j){
9         System.out.println("Method dengan 2 parameter = "+i+ " & "+j);
10    }
11    public void Tampil(String str){
12        System.out.println(str);
13    }
14
15    public static void main(String a[]){
16        Overloading objek = new Overloading();
17        objek.Tampil();
18        objek.Tampil(8);
19        objek.Tampil(6,7);
20        objek.Tampil("Hello world");
21    }
22 }

```

Overriding method adalah kemampuan dari subclass untuk memodifikasi method dari superclass-nya, yaitu dengan cara menumpuk (mendefinisikan kembali) method superclass-nya. Contoh overriding method dapat dilihat pada subclass “Mobil” yang mendefinisikan kembali method keterangan() dan hapus() dari class “Kendaraan”.